Engineering Technical Directive **No. 04-006**

Subject: Creating and Using Compression Algorithms for Data Pool Insert 06-May-04
The following directive is issued to all DAACS and labs.

Issue: With Release 7, a new Data Pool capability is provided which allows a DAAC the option of compressing Data Pool files at insert time, using a compression algorithm chosen and provided by the DAAC. (Reference Ticket DP_S4_07: Support Compression on Data Pool Insert). When creating and using compression algorithms and corresponding decompression algorithms for this capability, the DAAC must comply with the requirements and instructions defined in this directive. Note that the capability is limited to science files, i.e., XML and browse files are not compressed.

Fix: Document the requirements for creation of DAAC-provided compression algorithms and corresponding decompression algorithms, and the instructions for setting up compression on Data Pool insert using the Data Pool Maintenance GUI.

Implementation: Compression algorithms and corresponding decompression algorithms provided by the DAAC for use with the Compression on Data Pool Insert capability must comply with the requirements below. The Data Pool Maintenance GUI may be used to configure the Compression on Data Pool Insert capability, per instructions below.

1. Creating a compression algorithm

The compression algorithm must be created as a Unix command line sequence. The command line sequence may contain the name of a control file, which contains additional information needed by the compression algorithm. (e.g., in the case of HDF compression, a control file may be needed to indicate which objects in an HDF file to compress.)

Each compression algorithm provided must:

a) return an exit code of 0 for success and a non-zero exit code for failure;

NOTE: When providing the command line sequence for a compression algorithm, DAACs should recognize that a non-zero exit code from the compression algorithm would result in failure of the granule to be inserted into the Data Pool. Therefore, consideration should be given to desired compression algorithm behavior in non-nominal situations, such as where compression will not actually reduce the size of the input file. For example, for Unix compression, if the -f option is included in the command line sequence, this will force compression of the input file even if its size is not actually reduced, resulting in an exit code of 0 and a successful Data Pool insert for this case. If the -f option is not included in the command line sequence, a non-zero status code will be returned for input files, which do not actually reduce in size, and these files will not be inserted into the Data Pool.

b) be able to overwrite an existing compressed file (e.g., in the case where there is an error on the first attempt to insert the file in the Data Pool);

c) accept a %infile command line parameter indicating the name of the file to be compressed;

d) (optionally) append a default extension to the name of the compressed file;

e) place the compressed file in the same directory as the input file; and

f) export SANergy environment variables so that compression i/o occurs over the SANergy fabric (i.e., in a FUSED mode)

Subject: Creating and Using Compression Algorithms for Data Pool Insert 06-May-04

An example of a compression algorithm script file including SANergy environment variables is:

```
#! /bin/ksh

LD_PRELOAD=libSANergy.so

SANTISDIR=/opt/SANergy/nls/codeset

export LD_PRELOAD

export SANTISDIR


OTHER_ARGS=$*

 exec /bin/compress $OTHER_ARGS
```

NOTE: Due to an existing problem with the SANergy COTS software (see NCR ECSed40346: fused gzip gives bad gz files), gzip and unix compression command line sequences must include an intermediate step of writing the compressed file to standard out before renaming it with a compression extension. An example is:

```
/bin/gzip -1 -c $1 > $1.gz && /bin/rm -f $1
```

2. Creating a decompression algorithm

The decompression algorithm must be created as a Unix command line sequence.

Each decompression algorithm provided must:

a) return an exit code of 0 for success and a non-zero exit code for failure;

b) accept a %infile command line parameter indicating the name of the file to be decompressed; and

c) place the decompressed file in the same directory as the input file.

3. Storing the compression and decompression algorithms

Each compression algorithm must be set up such that it can be executed by the Data Pool Insert Utility on the x0dps01 host (i.e., must be placed in a directory which is visible to the Data Pool Insert Utility, and must have at least execute permissions for the UNIX user account running the Data Pool Action Driver (cmshared, allmode). Each corresponding decompression algorithm must be set up such that it can be invoked by the HEG Front End on the x0dps01 host. (The full path names for the compression algorithm and corresponding decompression algorithm are specified by the DAAC when defining compression algorithms using the Data Pool Maintenance GUI. The DAAC must also ensure that any supporting files which the compression or decompression algorithms require are accessible.)

4. Setting up Compression on Insert using the Data Pool Maintenance GUI

   a. Define new compression/decompression algorithms using the Compression Algorithms tab, and the Add Compression Algorithm link. Assign a unique label (up to 10 characters) to the compression algorithm.

Subject: Creating and Using Compression Algorithms for Data Pool Insert 06-May-04

b. Associate a compression algorithm with a Data Pool collection, using the Collection Groups tab. Click on the link for the appropriate collection group, and then on the List of Collections table, click on the link for the appropriate collection. On the Detail Information page, click on the Modify Collection link, and then choose a Compression Command Label from the pull-down list.

c. Turn Data Pool insert compression on and off at the system level, by choosing the appropriate setting of the CompressOnInsert parameter on the Configuration Parameters tab.

5. Monitoring Data Pool insert performance when compression is turned on

Insert performance may be monitored using the Data Pool Insert Utility logs and SAR data on the x0dps01 host. The extent of compression usage (at the collection level or at a Data Pool-wide level) should be adjusted according to DAAC Data Pool insert performance goals. Note that in the case of nonECS data, if DAAC monitoring shows that compression on insert is too costly performance-wise, the nonECS data files may instead be pre-compressed using the same compression algorithm before inserting them into the Data Pool.

The DAAC can obtain information on the timings and effectiveness of compression by querying (via SQL) the DlFiles table. It contains for each compressed file, the original and compressed size, as well as the compression time. Note that the time that is stored represents the *elapsed* time, not the CPU time.

Point of Contact: Kathleen L. Carr, email: kcarr@eos.hitc.com
Phone: 301/925-0642

Approved By: Mary Armstrong /s/ 05/07/04
IPT Manager, EMD Sustaining Engineering

Reference CCR: 04-0236

--------End of Directive-------